App.js:

import React from "react";

import { ListProvider } from "./ListContext";

import AddItem from "./AddItem";

import ItemList from "./ItemList";

import "./styles.css";

const App = () => {

  return (

    <ListProvider>

      <div className="app-container">

        <h1>Item Management</h1>

        <AddItem />

        <ItemList />

      </div>

    </ListProvider>

  );

};

export default App;

styles.css:

.app-container {

    text-align: center;

    font-family: Arial, sans-serif;

    padding: 20px;

    background-color: #f4f4f4;

    min-height: 100vh;

  }

  .add-item-container {

    margin-bottom: 20px;

  }

  .input-field {

    padding: 8px;

    margin-right: 10px;

    border: 1px solid #ccc;

    border-radius: 4px;

  }

  .add-button {

    padding: 8px 12px;

    background-color: #28a745;

    color: white;

    border: none;

    border-radius: 4px;

    cursor: pointer;

  }

  .add-button:hover {

    background-color: #218838;

  }

  .item-list-container {

    background: white;

    padding: 20px;

    border-radius: 8px;

    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

    max-width: 400px;

    margin: 0 auto;

  }

  .item-list {

    list-style: none;

    padding: 0;

  }

  .item {

    display: flex;

    justify-content: space-between;

    padding: 10px;

    border-bottom: 1px solid #ddd;

  }

  .remove-button {

    background-color: #dc3545;

    color: white;

    border: none;

    padding: 5px 10px;

    border-radius: 4px;

    cursor: pointer;

  }

  .remove-button:hover {

    background-color: #c82333;

  }

ListContext.js:

import React, { createContext, useReducer } from "react";

const ListContext = createContext();

const listReducer = (state, action) => {

  switch (action.type) {

    case "ADD\_ITEM":

      return [...state, { id: Date.now(), name: action.payload }];

    case "REMOVE\_ITEM":

      return state.filter(item => item.id !== action.payload);

    default:

      return state;

  }

};

const ListProvider = ({ children }) => {

  const [list, dispatch] = useReducer(listReducer, []);

  return (

    <ListContext.Provider value={{ list, dispatch }}>

      {children}

    </ListContext.Provider>

  );

};

export { ListContext, ListProvider };

AddItems.js:

import React, { useState, useContext } from "react";

import { ListContext } from "./ListContext";

import "./styles.css";

const AddItem = () => {

  const [item, setItem] = useState("");

  const { dispatch } = useContext(ListContext);

  const addItem = () => {

    if (item.trim() !== "") {

      dispatch({ type: "ADD\_ITEM", payload: item });

      setItem("");

    }

  };

  return (

    <div className="add-item-container">

      <input

        type="text"

        value={item}

        onChange={(e) => setItem(e.target.value)}

        placeholder="Enter item"

        className="input-field"

      />

      <button className="add-button" onClick={addItem}>Add Item</button>

    </div>

  );

};

export default AddItem;

ItemList.js:

import React, { useContext, useMemo } from "react";

import { ListContext } from "./ListContext";

import "./styles.css";

const ItemList = () => {

  const { list, dispatch } = useContext(ListContext);

  const memoizedList = useMemo(() => list, [list]);

  return (

    <div className="item-list-container">

      <h3>Item List ({memoizedList.length})</h3>

      <ul className="item-list">

        {memoizedList.map((item) => (

          <li key={item.id} className="item">

            {item.name}

            <button className="remove-button" onClick={() => dispatch({ type: "REMOVE\_ITEM", payload: item.id })}>Remove</button>

          </li>

        ))}

      </ul>

    </div>

  );

};

export default ItemList;